

REMARKS

In an office action dated 28 December 2001, the Examiner objects to the specification and the claims. Furthermore, the Examiner rejects claims 1-16 and 33-48. Applicants amend the specification and claims to overcome the objections and rejections. Applicants also respectfully traverse the rejection of claims 1-16 and 33-48. In light of the amendments and following remarks, Applicants request that the Examiner allow this application.

Applicants formally elect to prosecute claims 1-16 and 33-48 as per the telephone conversation of 8 November 2001. Claims 17-32 and 49-64 are withdrawn from consideration.

Applicant has amended to the specification to include the preliminary amendment of 2 July 1998. Applicant has also amended the specification to correct typographical and editorial errors in the specification. No new matter has been entered.

Applicant has corrected the antecedent errors cited in claims 1 and 33 to overcome the 35 USC 112 rejection of the Examiner. Therefore, the Applicants respectfully request that this rejection be removed.

The examiner rejects claim 1 under 35 USC 103(a) as being unpatentable over U.S. Patent Number 5,119,479, issued to Arai et al (Arai). Amended claim 1 recites storing input data in a data entity and storing identification information of the input device of the input data in the data entity. The data entity is then transferred to the software application. As described in the specification, the data entity is a data structure in which the data is stored and passed to the software application. Storing the data in the entity allows objects in object oriented programming or user defined data classes to be used to manipulate the data and transmit between device and the application. This allows a programmer to easily configure and change a system since the workings of the object in manipulating the data do not need to be known to the programmer.

Arai does not teach the storing of the data in an entity or a data structure. Aria teaches a system for providing a rule based system that using a name associated with an input device for determining the use of data receive from a device by an application. In Aria this is no mention of storing of the data in an entity and then transferring the entity to the application. The only teach of transferring of data with the name of the device and the rule that determines use of the data. (See Column 6, lines 24-49). Thus, Aria does not teach the storing of the data and identification in an entity as recited in Amended claim 1. For this reason, Applicants respectfully request that amended claim 1 be allowed.

Claims 2-16 are dependent from amended claim 1 and are allowable for at least the same reason as amended claim 1. Other arguments as to the allowability of claims 2-16 are moot and omitted for brevity. For these reasons, Applicants request claims 2-16 be allowed.

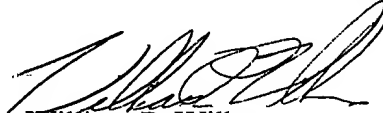
Claim 33 recites a computer which embodies components which perform the method of amended claim 1 and is allowable for at least the same reason as amended claim 1. For this reason, applicants respectfully request claim 33 be allowed.

Claims 34- 48 are dependent upon amended claim 33. Therefore, claims 34-48 are allowable for at least the same reasons as amended claim 33. . Other arguments as to the allowability of claims 34-48 are moot and omitted for brevity. For these reasons, Applicants request claims 34-48 be allowed.

If the Examiner has any question regarding this amendment or any other aspect of this application, the Examiner is respectfully requested to telephone the undersigned at 775-586-9500.

Respectfully submitted,  
SIERRA PATENT GROUP, LTD.

Dated: *28 March 2002*

  
William P. Wilbar  
Reg. No. 43,265

Sierra Patent Group, Ltd.  
P.O. BOX 6149  
Stateline, NV 89449  
(775) 586-9500

## APPENDIX – MARKED UP AMENDMENTS

In the Specification

Paragraph beginning at page 1, line 8, has been amended as follows:

--This application is related to [application Serial No. \_\_\_\_] U.S. Patent Number 6,237,053 B1, filed on the same day as [the present] this application, entitled "CONFIGURABLE OPERATING SYSTEM FOR I/O CONNECTIVITY"[,Attorney's Docket No. SYM-0625-1].

Paragraph beginning at page 2, line 23 been amended as follows:

Application developers implement their programs by selecting controls from a menu of control systems and placing each control in the desired location on the form. The properties associated with the control type [select] selected are then defined by the developer in order for the control to behave in the desired fashion. Each control type is implemented by a code module [that] which defines the behavior for that control type. The behavior for the control can be adjusted by setting the values of the properties defined for that control type. For example, the behavior for a text box control type accepts data input into its associated field from the keyboard.

Paragraph beginning at Page 6, line 12, has been amended as follows:

Moreover, the invention allows a robust integration of data between application programs and the operating system. The user no longer needs to physically perform the data transfer function or to develop custom handler software for the data. Once specified in a form, [date] data is dynamically placed into its destination field on the form. Moreover, the invention performs the data placement without requiring rigid rules and custom controls for each field.

Paragraph beginning at page 12, line 13, has been amended as follows:

The data output objects within [the] object class 104 operate in a similar manner. The core object 108 [provide] provides an output 120 in a defined format. Depending upon the output device which the program needs to use, or which is selected by the user, the program directs the output to an appropriate output object, which converts it into a format understood by the corresponding peripheral.

Paragraph beginning at page 20, line 8, has been amended as follows:

--Referring now to Figure 7, one implementation of an Input Requestor process 500 is illustrated. First the process 500 constructs the device using InputDevice.InputDevice (step 502). Next, the process 500 constructs a controller using an InputController.InputController function and passing it the device information (step [405] 504). The process 500 then enables the controller using an InputController.Enable function step (506) and obtains a buffer using an InputController.MakeBuffer function (step 508).--

Paragraph beginning at page 20, line 19, has been amended as follows:

Next, the process 500 determines whether the buffer is ready using an InputBuffer.GetReady function (step 514). From step 514, the process 500 determines whether the buffer is ready in step 516. If not, the process loops back to step 514 until the buffer is ready. From step 516, when a buffer becomes available, the process of Figure [9] 5 proceeds to step 518 where it obtains data from the buffer using an InputBuffer.GetString function. Further, the data is processed in the buffer (step 520). The process of Figure [9] 5 then checks whether additional input is needed (step 522). If so, the process loops back to step 512 to continue requesting input into the buffer. Alternately, the process of Figure 5 exits (step 524).

In the Claims:

1. (Amended) A method for transferring data from an input device to a software application having one or more data field, including the steps of:

storing [the] data from an input device in an entity;  
storing identification information regarding the input [source] device of the data in said entity;  
transferring said entity to the software application;  
associating said entity with [one of said data fields] a data field in the software application based on said identification information.

Claims 17-32 are canceled.

33. (Amended) A computer system for transferring data from an input device to a software application having one or more data fields, including:

a memory writer which stores the data from the input device in an entity and stores identification information regarding the input [source] device of the data in said entity;  
a sender which transfers said entity to the software application; and  
a matcher which associates said entity with [one of said data fields] a data field in the software application based on said identification information.

34. (Amended) The computer system of claim 33 further including [a]:  
an entity modifier which forms a data object of said entity.

Claims 49-64 are canceled.